Available on CMS information server

CMS NOTE 2007/007

The Compact Muon Solenoid Experiment

CMS Note

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

March 21, 2007

A Linear Algorithm for the CMS Muon Drift Tubes Local Pattern Recognition

A. De Min

EPFL, Lausanne, [a] *and INFN, Sez. Milano-Bicocca*

Abstract

In this note a new, linear reconstruction algorithm is suggested for the CMS barrel muon chambers that detects the presence of a muon track, performs optimal pattern recognition, solves left-right ambiguities and fits the muon track segment within each chamber with a simple single-step procedure. The algorithm uses mixed-integer programming techniques developed in operations research. Compared to the sequential reconstruction method presently used it is potentially unbiased, more reliable and possibly faster.

[a] On leave from *Politecnico di Milano*

1 Introduction

The CMS barrel muon system consists of a modular structure of drift tubes with rectangular profile and central anode wire. The cells are grouped in sets of staggered layers, the so called *muon chambers*, positioned in regions where the magnetic field is relatively low so that muon trajectories are expected to be approximately straight lines. The excellent uniformity of the electric field in the electron drift region guarantees a spatial resolution of 250 μm [1].

In the present version of the CMS reconstruction software muon segments are locally built inside each DT chamber by means of a rather lengthy iterative procedure. In a first step pairs of hits belonging to distant layers are connected through segments that are then extended using additional hits. In a second step the best track segment candidates are selected based on χ^2 and hit multiplicity criteria, while ghosts and left-right ambiguities are removed. Finally the hit positions are refitted using the segment information and the segments themselves are eventually refitted. Line segments are reconstructed separately in the RZ and Rϕ projections and are joined together only at the end of the local pattern recognition. A more detailed description of the algorithm currently in use can be found in [2].

In this note a new method is suggested for local muon-segment reconstruction within each CMS muon chamber based on *linear programming* [3], a branch of mathematics developed in the second half of the last century to solve minimization problems related to economy, finance and logistics. Compared to the present approach the proposed method is simpler, potentially faster and, which is most important, should be free from any possible bias deriving from the initial *seed* (such as a pair of hits or a triplet of aligned points, for example) used to initiate the segment reconstruction. In the past similar algorithms were applied, for example, to vertex detector hits to obtain optimal track reconstruction in regions close to the primary vertex [4].

In order to illustrate these concepts, in the next section an algorithm based on a set of linear inequalities is proposed that reconstructs straight lines in a plane in the presence of spurious hits. Its application to the CMS muon system will be discussed in the following section.

It should be emphasized that the technique here suggested does not apply uniquely to straight-line fits but can be used for arbitrary parametric curves with *linear dependence on the parameters*. However, since the effectiveness of the method critically depends on the number of noise hits, here we only consider its use in the relatively clean environment of the CMS muon chambers, while for detectors with larger hit densities such as the CMS tracker further studies are required to verify its applicability.

2 Straight-line reconstruction in two dimensions in the presence of noise

In this section we will consider the problem of selecting, out of a set of several points in a plane, the subset of points that are approximately disposed in a straight–line configuration and obtaining the corresponding best straight–line fit. The problem is equivalent to performing a straight–line trajectory search and fit in a bi–dimensional readout detector producing discrete points, some of which are truly aligned while others are randomly distributed (as a result of noise). Such is the case of the CMS muon chambers. We will also assume, as is true in the CMS DT muon chambers, that one of the two coordinates of each point is known with absolute precision, say the Z coordinate, while the X coordinate (the drift direction in the CMS muon cells) is affected by an experimental uncertainty that may depend on the considered point. We will show how the problem can be formulated in linear terms and consequently solved using linear programming techniques.

2.1 Linear straight-line fit

If we write the straight-line equation as:

$$X = mZ + q, \tag{1}$$

then the hit P_i of coordinates (Z_i,X_i) should be associated with the straight line if its coordinates satisfy the constraint:

$$|mZ_i + q - X_i| \leq \Delta_i, \tag{2}$$

where Δ_i is an appropriate resolution parameter (that might depend on the point), while it should be discarded otherwise. The Δ_i parameter is a sort of maximum tolerance for the point mis-measurement and, in practice, should be chosen between three and five times the position standard deviation. Strictly speaking the above relation is nonlinear due to the presence of the absolute value. However the constraint expressed by inequality (2) can be re-formulated in linear terms by doubling the number of inequalities, or *constraints*, (which have to be simultaneously satisfied):

$$\begin{aligned} mZ_i + q - X_i - \Delta_i &\leq 0 \\ -mZ_i - q + X_i - \Delta_i &\leq 0 \end{aligned} \tag{3}$$

In the case where one has a set of N experimental points a *linear* fit can be performed on *all points* by minimizing the linear (weighted) sum of the residuals σ_i with respect to a straight line of unknown equation. The problem is then formulated as follows:

$$\min \sum_i \sigma_i/\Delta_i \tag{4}$$

$$\begin{aligned} \text{subject to } (1 \leq i \leq N): & \\ mZ_i + q - X_i - \sigma_i &\leq 0 \\ -mZ_i - q + X_i - \sigma_i &\leq 0 \\ m, q, \sigma_i &\geq 0 \end{aligned} \tag{5}$$

One clear advantage of this approach compared to the classical least-squares fit is that it de-emphasizes the role played by distant points (i.e. points at the limit of compatibility), as it involves variables raised to the first rather than the second power. The minimization of the objective function $\sum_i \sigma_i/\Delta_i$ under linear constraints can be performed in an extremely fast way using, for example, the simplex method [5], one of the milestones of linear programming, although alternative and equally effective methods, such as the more recent *interior points method* [6], are also available. The condition $m, q \geq 0$ is not in the original problem but it is necessary for the application of linear programming techniques. As we will see in the following, this additional constraint does not represent a limitation nor a loss of generality.

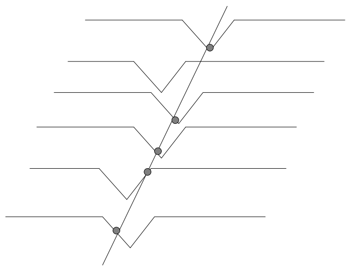

Figure 1: Visualization of the fitting process as the minimization of the sum of piecewise linear potential functions. The positions of those points that are incompatible with the fitted straight line are irrelevant in the minimization.

2.2 Straight-line pattern recognition

The formulation of the problem described in (4) and (5) uses all points to define the optimal straight-line parameters. In other words it assumes that, in the presence of spurious hits, the pattern-recognition step, i.e. the exclusion of those points (outliers) for which $\sigma_i > \Delta_i$ (or equivalently, that do not satisfy the inequalities (3)), has already been performed. In the following we will show how the pattern recognition step can also be expressed in a linear formalism and included directly in the linear fit procedure.

The objective is then to reformulate the minimization problem (4) in such a way that the constraints (5) are applied only to those points that should be included in the fit. This can be done by introducing a mechanism that automatically relaxes these constraints for the outliers. Such a mechanism could be visualized as the minimization of a sum of piecewise linear potential functions that are completely flat far from the observed hits and have dips (of width $2\Delta_i$) with straight walls at 45^o inclination correspondning to hit positions, as represented in Figure 1. Once the distance of a hit from the line exceeds the resolution parameter that hit loses any influence on the minimum of the potential.

2.3 Linear formulation of the global straight-line reconstruction problem

In order to use linear programming to solve the pattern recognition problem defined by the previous section, one has to demonstrate that it can indeed be formulated in linear terms.

In fact the *linearization* of the pattern recognition and fitting problems can be achieved by means of the following compact formulation:

$$\begin{aligned}
\min \sum_i \sigma_i/\Delta_i & & \qquad (6)\\
\text{subject to } (1 \le i \le N): & & \\
mZ_i + q - X_i &\le \sigma_i + \lambda_i M \qquad (7)\\
-mZ_i - q + X_i &\le \sigma_i + \lambda_i M \qquad (8)\\
\sigma_i - \Delta_i &\le \lambda_i M \qquad (9)\\
-\sigma_i + \Delta_i &\le (1-\lambda_i)M \qquad (10)\\
m, q, \sigma_i &\ge 0\\
\lambda_i &= \{0, 1\},
\end{aligned}$$

where M $\gg 1$ is a large, arbitrary, fixed real number and λ_i are a set of additional binary variables (one per candidate point). Here the change of slope in the single-point potentials is obtained by means of the λ_i variables, in association with the M constant defined above. This technique, known as *big-M method*, is frequently used in operations research to linearize and rapidly solve practical problems related, for example, to logistics where one needs to reproduce the effects of fixed costs.

With this formulation the positions of those points for which $\sigma_i > \Delta_i$ become irrelevant for the minimization process because the constraints (7) and (8) are always satisfied and the σ_i variables are limited from below by the corresponding Δ_i terms.

This can be formally proven as follows, given that λ_i can only assume the 0 and 1 values:

1. If $\lambda_i = 0$ then from the constraint (9) one obtains $\sigma_i \le \Delta_i$ and the point is considered in the fit due to the relations (7) and (8).
2. In the opposite case that $\lambda_i = 1$ then from the constraint (10) one obtains $\sigma_i \ge \Delta_i$ while the relations (7) and (8) become always true, implying that the point has no influence on the fit.

Viceversa:

1. If $\sigma_i \le \Delta_i$ then the constraint (9) is always satisfied, while the constraint (10) necessarily implies $\lambda_i = 0$.
2. In the opposite case that $\sigma_i > \Delta_i$ the constraint (9) implies $\lambda_i = 1$, while the relation (10) is always satisfied.

In conclusion, by simply using linear relations, a contribution for each hit to the objective function (6) is obtained which equals σ_i/Δ_i (≤ 1) when $\sigma_i \le \Delta_i$ and equals exactly 1 otherwise. In the solution vector, m and q will be the best-fit parameters, σ_i the hit residuals and λ_i will be 0 (1) if point i is (is not) included in the fit.

In practice, the method is illustrated (in an over-simplified case) by the fit of Figure 2. In the plot six hits are considered, each having the same (vertical) resolution parameter Δ_i of 0.1 units. The effect of noise is simulated by one hit at the top left corner of the figure. The classic least-squares (LS) straight-line fit blindly considers all hits and produces an incorrect result. The method proposed here (with an M parameter of 100), based on least absolute values (LAV), when applied to all hits automatically discards the noise and produces the correct result. In this specific example, the central point 4 with horizontal coordinate equal to 3.0 is also excluded by the fitting procedure as incompatible with the line equation. The calculated residuals σ_i and the corresponding λ_i values are described in Table1 that shows how both excluded hits (point 2 and 4) are assigned the same residual value $\sigma_i = \Delta_i$, irrespective of their distance from the final fit.

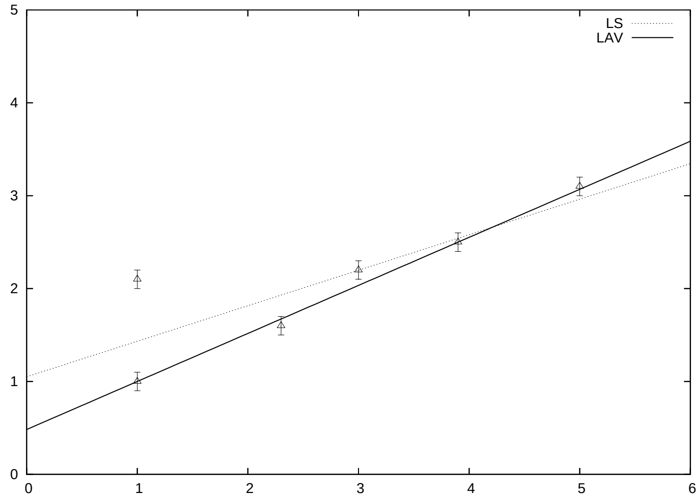

Figure 2: An illustration of the difference between the classical straight-line fit based on the least-squares (LS) method and the linear method (LAV) proposed in this note. The noisy hits at the top left corner and at the center of the plot are automatically ignored by the fit using integer programming techniques.

It should be noted that similarly to the least-squares method the minimization of the objective function (6) does not guarantee by itself that the straight-line fit passes near the largest set of hits, since what is strictly minimized is just the linear sum of residuals. In case this feature is considered to be desirable then the objective function in (6) should be modified as follows:

$$\min \sum_i (\sigma_i/\Delta_i + \lambda_i M). \tag{11}$$

Assuming M to be much larger than the number of points, the algorithm will *first* determine the set of straight-line equations that are compatible with the largest number of points and *then*, among all such solutions it will select the straight-line equation with the lowest sum of residuals.

To summarize, by solving the minimization problem in (6) or (11) one immediately obtains the parameters of the best straight-line fit, the explicit list of the hits included in the fit (via the λ_i variables) and the corresponding residuals σ_i.

2.4 Mixed-integer programming

Since some of the unknowns of the problem are now binary, one normally speaks of *mixed-integer programming* (MIP) rather than *linear programming* (LP) in this case. Mixed-integer problems cannot be solved in a straightforward way with methods such as the simplex, since the need for integer solutions imposes the use of additional iterative procedures. For example one widely used method to obtain integer solutions is the so called *branch-and-bound* technique that consists in adding sequentially new constraints (following a sort of *tree* structure) imposing integer values on non-integer variables and solving the corresponding linear problem. Alternatively, methods based on the sequential introduction of *cutting planes*, with the aim of limiting the space of the feasible solutions to integer values, may also be applied.

Table 1: The values of X_i, Y_i, σ_i and λ_i of the six points considered in the example described in the text. All points have a Δ_i of 0.1. The M-value is 100.

Point	X_i	Y_i	σ_i	λ_i
1	1.0	1.0	0.0	0
2	1.0	2.1	0.1	1
3	2.3	1.6	0.072	0
4	3.0	2.2	0.1	1
5	3.9	2.5	0.0	0
6	5.0	3.1	0.031	0

One may think that this complication would spoil the advantage of the linear approach with respect to the classical combinatorial procedure, but in general this is not the case. In fact the solution of the proposed linear method is expected to be faster than traditional sequential methods given that:

1. mixed-integer linear problems can be solved by specialized software tools that are extremely fast;
2. most of the λ_i variables will have already binary values after the first iteration and only some ambiguities will need to be treated explicitly;
3. solvers adopt methods to avoid explicit enumeration of *all* combinatorial possibilities by using *implicit enumeration* techniques to reduce efficiently the space of possible solutions.

Several solvers (libraries) are available on the market which are specialized for mixed-integer problems. Among commercial solvers, the most widely used are ILOG/Cplex [7], Xpress-MP [8] and the IBM/OSL [9] solver. These software products are relatively expensive in general terms, but special agreements are always possible in case of research projects.

An alternative possibility consists in using open source packages, that, apart from being completely free, have the non-negligible advantage of being available in (C++) source code. One of the best candidates on the market is the CBC-CLP solver for mixed integer problems (that was used here to obtain the fit in Figure 2), downloadable from the COIN-OR (*Common Initiative for Operation Research*) web site [10] and usable under the OSI [11] licence. Alternatively the less sophisticated library provided by GLPK (*GNU Linear Programming Kit*) [12] may also be used.

2.5 Multi-track pattern recognition

In the previous section we showed how a pattern recognition problem can be formulated in linear terms and how linear programming techniques can be exploited to obtain a fast linear fit with the exclusion of the outliers. However in the case considered above only one track was assumed to be present. In this section we will consider the immediate generalization of the problem, where several tracks could be simultaneously present. In this case the problem consists in determining the smallest number of subsets of aligned points (technically called *feasible subsystems*) which would constitute a *partition* of the total set. Problems of this kind are known as MIN-FS (*minimum set of feasible subsystems*) and have been widely studied in the literature as they appear in many disciplines, from digital image reconstruction to social sciences [13]. In this context, however, we will only consider the simplest sequential approach, which consists in applying the following *greedy* procedure, aimed at determining, at each iteration, the *maximum feasible subsystem*:

1. From the initial set of points, the best straight-line fit is determined using the method described in the previous sections.
2. The straight-line parameters are stored (as a first candidate track) and the corresponding points are removed from the initial set.
3. The procedure is recursively applied until the last reconstructed straight line contains an insufficient number of points, according to some pre-defined quality criterion.

This approach is certainly justified for the specific case of the CMS muon chambers given the excellent spatial resolution and the low track density, as the probability for a hit to be shared between two or more tracks is negligible.

3 Application to the CMS barrel muon chambers

Compared to the general case presented above, the CMS muon system has only one additional complication due to the presence of left-right ambiguities in the drift chambers. This implies that for each drift tube two candidate hits should be considered, symmetrically positioned with respect to the anode wire.

The simplest solution is to consider both candidate hits as independent points and then add, for each couple of hits i and j associated to the same drift cell, the linear constraint:

$$\lambda_i + \lambda_j \geq 1. \tag{12}$$

In fact the variable λ_i is 0 when the corresponding point is used in the fit and 1 otherwise. The constraint in (12) implies that at most one of the two hits is included in the fit. However this implies that the number of constraints is increased by the number of hit pairs considered, which will certainly slow down the algorithm. Since in general we do not expect track segments to be associated to more than one hit per layer, the above set of constraints can be replaced by a limited number of constraints, one per layer, stating that no more than one hit per layer should be considered in the fit:

$$\forall \text{ layer j } \sum_{\mathrm{i}} \lambda_{\mathrm{ij}} \geq (\mathrm{N_j} - 1), \tag{13}$$

where the index j runs over the layers, λ_{ij} is the binary variable associated to hit i in layer j and $\mathrm{N_j}$ is the number of hits in layer j (i.e. twice the number of hit pairs).

As already mentioned the constraints m,q≥ 0 do not represent a real difficulty. In fact it is always possible to express any relative number as the difference between two arbitrary *positive* numbers. It is thus sufficient to express the straight-line coefficients as m = (m$^+$ - m$^-$) and q = (q$^+$-q$^-$), with m^+, m^-, q^+, $\mathrm{q}^- \geq 0$ in (6-10) to generalize the problem. By simply limiting the range of the m and q parameters, the user can also impose additional constraints on the straight-line angular acceptance.

Apart from the above modifications, the formalism outlined in Section 2 is applicable in a straightforward way to the CMS muon system, where the drift direction can be identified with the X coordinate and the perpendicular direction with Z, if one considers the Rϕ and RZ views separately, although the same formalism could be applied in three dimensions, with an appropriate choice of the various resolution parameters.

It should also be emphasized that in case of the number of points in the fit is precisely three, the result of the fit might seem surprising at first sight, since the fitted straight line would always pass through the extreme points (given that any alternative solution would inevitably increase the linear sum of the hit residuals). For a larger set of points the result becomes increasingly close to the least-squares fit. If this feature is considered unpleasant one should use the MIP programming approach presented here only for pattern recognition and then fit the selected points analytically with the least-squares method.

The advantages of a reconstruction method based on linear programming with respect to the formulation presently implemented in the CMS software could be listed as follows.

1. The reconstruction would be rigorously unbiased, since it does not require the determination of any initial *seed* (such as a triplet of aligned points) for the reconstruction of track segments.

2. It would also be more general and *intuitive*, i.e. more similar to the (global) approach followed by the human brain in detecting geometrical patterns which appear *at first glance* rather than as a result of a sequential association process.

3. It could be significantly faster than conventional sequential methods, although this strongly depends on the size of the problem, i.e. on the number of tracks and of noise hits. For this reason its utilization in high-level trigger applications could be particularly relevant. In this case, for example, it would

be sufficient to relax significantly the resolution parameters Δ_i to detect *immediately* the presence of approximate linear patterns signalling the existence of a muon track segment.

4. It should be more stable since, opposite to least-squares fits, it weights solutions according to residuals raised to the first rather than the second power, which reduces the weight of hits with residual close to their Δ parameter.

5. Constraints on the straight-line angular acceptance can be imposed trivially by adding new (linear) constraints on the values of the m and q parameters as additional lines in (6).

6. It would be more *elegant*, as it can unify in a single procedure the pattern recognition and the segment fitting steps.

4 Conclusions

This note shows how the problem of detecting and reconstructing muon track segments within each CMS barrel muon chamber can be formulated in linear terms and how linear and integer programming techniques can be applied to solve it. The algorithm proposed provides optimal pattern recognition and straight-line fitting in a unified, global approach. The method is simple, reliable and unbiased and can be significantly faster than the conventional sequential approach. For these reasons it could find interesting applications in off-line reconstruction packages as well as in high-level trigger applications.

5 Acknowledgements

The author would like to address special thanks to Prof. Michel Bierlaire, *Transport and Mobility Laboratory*, EPFL, Lausanne, for his valuable comments on the draft and on the mathematical approach.

References

[1] CMS Collaboration, *The Muon Project Technical Design Report*, CERN/LHCC 97-32, CMS TDR 3.

[2] CMS Collaboration, PTDR Vol. I, *Detector Performance and Software*, CERN/LHCC 2006-001, CMS TDR 8.1.

[3] See, for example, G. B. Dantzig, *Linear programming applications and extensions*, Princeton University Press, 1998.

[4] J-F. Pusztaszeri, P.E. Rensing, T.M.Liebling, *Tracking elementary particles near their primary vertex: a combinatorial approach*, J. Global Optim. 16 (1995) 422-431.

[5] G.B.Dantzig, *Maximization of a linear function of variables subject to linear inequalities*, in T.C.Koopmans (ed.), *Activity analysis of production and allocation*, Johon Wiley & Sons, Inc, New York, 1951, pp 339347.

[6] G.L. Nemhauser and L.A. Wolsey, *Integer and combinatorial optimization*, Wiley, 1988. H.M. Salkin and K. Mathur, *Foundations of Integer Programming*, North-Holland, 1989.

[7] See `www.ilog.com`.

[8] See `www.dashoptimization.com`.

[9] See `www.research.ibm.com/math/OpResearch/adv_math_software.html`.

[10] See `www.coin-or.org`.

[11] See `www.opensource.org/`

[12] See `www.gnu.org/software/glpk/`

[13] For a review, see E. Amaldi, *Infeasible linear systems: Finding minimum partitions into feasible subsystems*, 3rd Workshop on Combinatorial Optimization, Aussois, France, march 1999.